

CoE197s

Sean G. Cabalse

2015-04566

Exercise 3: Building Images

1) Getting Setup

a) Removing old containers and Downloading new ubuntu:16.04

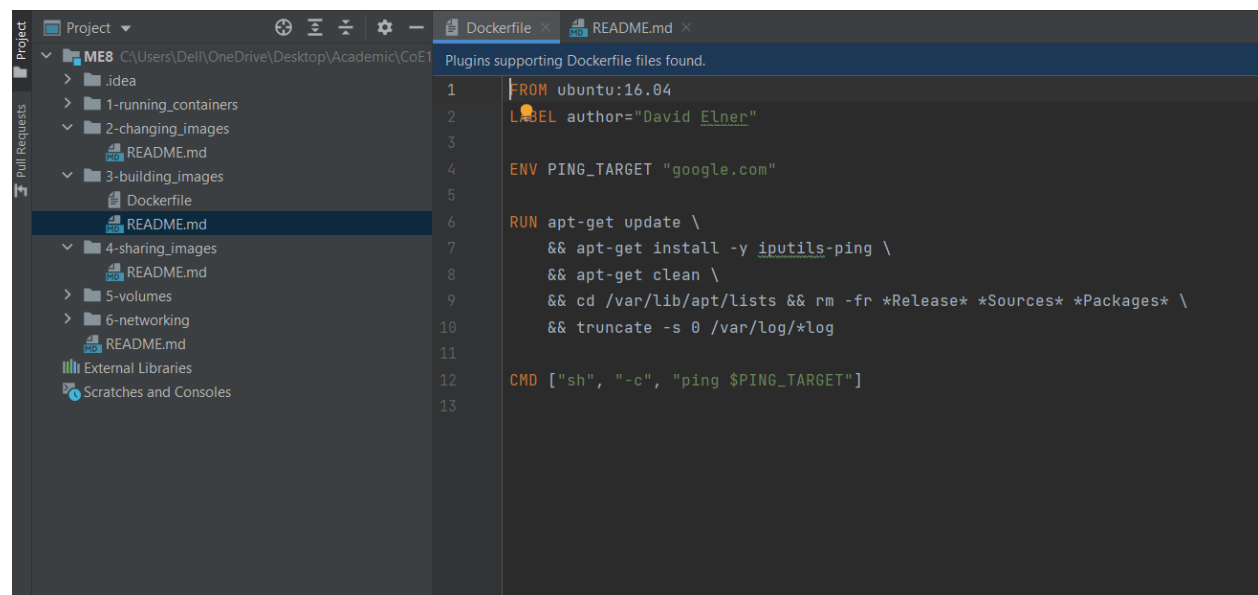
```
C:\Users\Dell\OneDrive\Desktop\Academic\CoE197s\ME8\1-running_containers>docker pull ubuntu:16.04
16.04: Pulling from library/ubuntu
Digest: sha256:eed7e1076bbcf342c4474c718e5438af4784f59a4e88ad687dbb98483b59ee4
Status: Image is up to date for ubuntu:16.04
docker.io/library/ubuntu:16.04

C:\Users\Dell\OneDrive\Desktop\Academic\CoE197s\ME8\1-running_containers>docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
seancabalse/ping    latest             9cd101b5af28       10 minutes ago     169MB
docker101tutorial   latest             ed92b27e7580       2 hours ago        28MB
seancabalse/docker101tutorial   latest             ed92b27e7580       2 hours ago        28MB
alpine/git           latest             2abeafdf5b9        3 days ago         25.1MB
ubuntu               16.04              aefd7f02ae24       3 weeks ago        134MB

C:\Users\Dell\OneDrive\Desktop\Academic\CoE197s\ME8\1-running_containers>docker rmi 9cd101b5af28
Untagged: seancabalse/ping:latest
Deleted: sha256:9cd101b5af28dc44fc5510eb9b1f4f511d7b2120b2b4d5fb2c289cfe429f8e90
Deleted: sha256:cf2446a31db1960a8934c8d2d454af3e407e23789eada27f66f17244d31f22d3

C:\Users\Dell\OneDrive\Desktop\Academic\CoE197s\ME8\1-running_containers>docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
docker101tutorial   latest             ed92b27e7580       2 hours ago        28MB
seancabalse/docker101tutorial   latest             ed92b27e7580       2 hours ago        28MB
alpine/git           latest             2abeafdf5b9        3 days ago         25.1MB
ubuntu               16.04              aefd7f02ae24       3 weeks ago        134MB
```

2) Creating a Docker File



```
1 FROM ubuntu:16.04
2 LABEL author="David Elner"
3
4 ENV PING_TARGET "google.com"
5
6 RUN apt-get update \
7     && apt-get install -y iputils-ping \
8     && apt-get clean \
9     && cd /var/lib/apt/lists && rm -fr *Release* *Sources* *Packages* \
10    && truncate -s 0 /var/log/*log
11
12 CMD ["sh", "-c", "ping $PING_TARGET"]
13
```

3) Building, Optimizing and Using Other Directives on the Docker File

```
C:\Users\Dell\OneDrive\Desktop\Academic\CoE197a\ME8\3-building_images>docker build -t seancabalse/ping .
[+] Building 133.7s (6/6) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring Dockerfile: 356B
=> [internal] load dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/ubuntu:16.04
=> [1/2] FROM docker.io/library/ubuntu:16.04
=> [2/2] RUN apt-get update && apt-get install -y iputils-ping && apt-get clean && cd /var/lib/apt/lists && rm -rf 'Release' 'Sources' 'Packages'
=> exporting image
=> => exporting layers
=> => writing image sha256:ce8f1ed892a65ebc9b53b9b127fb139f05aea03f62b46df7793e66aea316cbc
=> => naming to docker.io/seancabalse/ping

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them

C:\Users\Dell\OneDrive\Desktop\Academic\CoE197a\ME8\3-building_images>docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE	OS	ARCH
seancabalse/ping	latest	ce8f1ed892a6	34 seconds ago	130MB	linux	amd64
docker101tutorial	latest	ed92b27e7580	3 hours ago	28MB	linux	amd64
seancabalse/docker101tutorial	latest	ed92b27e7580	3 hours ago	28MB	linux	amd64
alpine/git	latest	2abeafdf5b9	3 days ago	25 MB	linux	amd64
ubuntu	16.04	aefd7f02ae24	3 weeks ago	134MB	linux	amd64

Optimizing the Dockerfile

Looking at new image, we can see it is 130MB in size versus its base image of 11MB. That's a pretty big change in size for installing some packages. Let's take a look at what apt-get does and see what we can do to push aspects of this image.

But why is it that much bigger? The secret is in the `rm` commands mentioned before, any filesystem changes are committed after the `rm` command completes. This includes any logs, or temporary data written to the filesystem which might be completely inconsequential to our image.

In our case, the use of `rm -rf` generates a lot of this fluff we don't need in our image. We'd need to modify those `rm` directives slightly.